

Revisión No Priorizada de Planes en Agentes Inteligentes

Gerardo Parra

Departamento de Ciencias de la Computación
UNIVERSIDAD NACIONAL DEL COMAHUE

Guillermo R. Simari

Departamento de Ciencias e Ingeniería de la Computación
UNIVERSIDAD NACIONAL DEL SUR

e-mail: gparra@uncoma.edu.ar, grs@cs.uns.edu.ar

Palabras Clave: INTELIGENCIA ARTIFICIAL, PLANEAMIENTO, DINÁMICA DE CREENCIAS.

Resumen

Los agentes inteligentes autónomos se ven obligados a modificar sus planes de acción debido al dinamismo del entorno en el que se desenvuelven. Nuestra línea de trabajo postula la conveniencia de adoptar el punto de vista del área de Dinámica de Creencias al considerar la actividad de replaneamiento de un agente inteligente. En trabajos previos hemos introducido un modelo para representar expansiones, contracciones y revisiones de grafos de planning. La operación de revisión ocurre cuando es necesario remover algunas piezas del plan global y reemplazarlas por subplanes alternativos que permitan llevar a cabo el plan de manera exitosa.

En este trabajo, introducimos la operación de revisión no priorizada de grafos de planning. Esta operación permite modelar la situación en la que un agente planificador recibe una nueva pieza del plan y decide incorporarla o no al grafo de planning.

1. Introducción

Los agentes inteligentes autónomos, debido a su proactividad, se ven obligados a considerar la satisfacción de sus metas a través de un conjunto estructurado de acciones que conforman un plan. El modelo BDI (*Belief, Desires and Intentions*)[GPP⁺99] para representar el conjunto cognitivo de un agente es una posibilidad interesante que permite estudiar el problema que introduce el dinamismo natural del entorno en el que un plan particular se desenvuelve. El entorno corriente, el mundo actual del agente, es representado con un modelo de creencias (*beliefs*) adecuado. Las metas del agente representan sus deseos (*desires*) y describen en forma parcial estados del entorno preferidos. Finalmente, los planes para alcanzar alguno de aquellos estados constituyen, en cierta forma, las intenciones (*intentions*).

El dinamismo del entorno provoca que algunos de los planes deban ser modificados para poder alcanzar las metas finales. Esta actividad de replaneamiento puede considerarse, una revisión del

mismo. Ciertas partes pueden ser conservadas, pero otras deben ser removidas y reemplazadas por subplanes convenientes que ofrezcan la posibilidad de éxito para el plan global.

Nuestra línea de trabajo postula la conveniencia de adoptar el punto de vista del área de Dinámica de Creencias[AGM85, Gär88, Han96b] al considerar la actividad de replaneamiento de un agente inteligente. En trabajos previos[PS01, PS02a, PS02b] hemos introducido un modelo para representar expansiones, contracciones y revisiones de grafos de planning.

Una de las características distintivas de la operación de revisión es la satisfacción del postulado de éxito. Es decir, el nuevo esquema de acción propuesto es incorporado al grafo de planning y, de esta manera, tiene prioridad sobre los esquemas ya existentes. En este trabajo, introducimos el operador de revisión no priorizada de grafos de planning. Esta operación permite modelar la situación en la que un agente planificador recibe una nueva pieza del plan y decide incorporarla o no al grafo de planning.

El artículo está organizado de la siguiente manera. En la sección 2 se presentan los principales conceptos relacionados al área de planning en Inteligencia Artificial. Se analiza un dispositivo de planning particular y se trata, en este contexto, la construcción de grafos de planning. En la siguiente sección presentamos los aspectos más relevantes relacionados con la dinámica de creencias. La sección 4 contiene las principales contribuciones de este trabajo. Se analiza la revisión de planes y se introduce la revisión no priorizada de grafos de planning. Finalmente, la sección 5 incluye las conclusiones del trabajo.

2. Planeamiento. Graphplan

El objetivo central del área de planning en el contexto de Inteligencia Artificial es construir algoritmos que hagan posible a un agente elaborar un curso de acción para lograr sus metas. El resultado producido por un dispositivo de planning es una secuencia de acciones las cuales, cuando son ejecutadas en un mundo que satisface la descripción del estado inicial, lograrán la obtención de la meta. Existe una amplia variedad de lenguajes para representar el mundo, las metas del agente y las acciones posibles. En este trabajo adoptamos la representación STRIPS[FN71] como lenguaje de representación.

La representación STRIPS describe el estado inicial del mundo mediante un conjunto completo de literales básicos (*ground*) y las metas son definidas como una conjunción proposicional. La teoría de dominio, es decir, la descripción formal de las acciones disponibles para el agente, completa la descripción del problema de planning.

Cada acción se describe mediante dos fórmulas: la precondition y el efecto o poscondición. Ambas están constituidas por una conjunción de literales y definen una función de transición de un mundo a otro. Una acción puede ser ejecutada en cualquier mundo w que satisfaga la fórmula de la precondition. El resultado de ejecutar una acción en un mundo w es especificado tomando la descripción de w , adicionando cada literal de la poscondición de la acción ejecutada y eliminando literales contradictorios.

Lo expuesto hasta aquí caracteriza un problema de planning clásico. Es importante destacar que

existen muchas suposiciones que simplifican el problema: tiempo atómico, no existen eventos exógenos, los efectos de las acciones son determinísticos, el agente es omnisciente, entre otras.

Uno de los dispositivos de planning más interesantes es Graphplan[BF95, BF97, Wel99]. El funcionamiento de Graphplan alterna entre dos fases: la construcción del grafo de planning y la extracción de la solución. La primera fase construye, a través de sucesivas etapas, un grafo de planning hacia adelante en el tiempo hasta que se logra una condición necesaria (pero que puede ser insuficiente) para la existencia de un plan. Luego, la fase de extracción de solución realiza un recorrido hacia atrás sobre el grafo, buscando un plan que resuelva el problema. Si no es hallada una solución, el ciclo se repite llevando a cabo una nueva etapa en la construcción del grafo de planning.

A continuación describimos, de manera resumida, cada una de las fases del funcionamiento de Graphplan. Una presentación detallada puede encontrarse en [Wel99].

2.1. Grafos de Planning

El grafo de planning contiene dos tipos de nodos: nodos de proposición y nodos de acción, organizados en niveles. Los niveles con numeración par contienen nodos de proposición (es decir, literales *ground*) y, en particular, el nivel cero consiste precisamente de las proposiciones que son verdaderas en el estado inicial del problema de planning. Los nodos presentes en niveles con numeración impar corresponden a instancias de acción. Existe uno de tales nodos por cada instancia de acción cuyas precondiciones estén presentes (y sean mutuamente consistentes) en el nivel previo.

Los nodos correspondientes a instancias de acción están conectados mediante arcos a los nodos de proposición (en el nivel anterior) que constituyen las precondiciones de la acción. Existen arcos adicionales (arcos de poscondición) que conectan los nodos de acción con los nodos de proposiciones (en el nivel siguiente) que se hacen verdaderas por efecto de la acción.

Es importante destacar que el grafo de planning representa acciones que pueden desarrollarse *en paralelo* en cada nivel de acciones. Sin embargo, el hecho que dos acciones estén presentes en el mismo nivel de un grafo de planning no significa que sea posible ejecutar ambas a la vez.

Central a la eficiencia de Graphplan es la inferencia de una relación binaria de exclusión mutua, denominada *mutex*, entre nodos presentes en el mismo nivel. La relación se define recursivamente como sigue:

- Dos instancias de acción en el nivel i son *mutex* si ocurre alguno de los casos siguientes:
 - el efecto de una acción es la negación del efecto de la otra (*Efectos Inconsistentes*)
 - el efecto de una acción elimina la precondición de la otra (*Interferencia*)
 - las acciones tienen precondiciones que son mutuamente exclusivas a nivel $i - 1$ (*Necesidades Conflictivas*).
- Dos proposiciones en el nivel i son *mutex* si una es la negación de la otra o si todas las formas de arribar a estas proposiciones (es decir, las acciones en el nivel $i - 1$) son *mutex* tomadas de a dos (*Soporte Inconsistente*).

A continuación presentamos, a modo de ejemplo, la especificación STRIPS del *dinner-date problem* [Wel99].

```
Condiciones Iniciales: (and (basura) (manosLimpias) (silencio))
Meta:                  (and (cena) (regalo) (not (basura)))
Acciones:

Cocinar                :precondition (manosLimpias)
                      :effect (cena)
Envolver               :precondition (silencio)
                      :effect (regalo)
Llevar_a_mano           :precondition
                      :effect and ( (not (basura)) (not (manosLimpias)))
Llevar_en_carretilla    :precondition
                      :effect and ( (not (basura)) (not (silencio)))
```

Figura 1: *Dinner-date problem*.

El *dinner-date problem* consiste en encontrar un plan para preparar una cita sorpresa para nuestra amada que se encuentra durmiendo. La meta del problema es sacar las bolsas de basura, preparar la cena y envolver el regalo. Existen cuatro acciones posibles: `cocinar`, `envolver`, `llevar_a_mano` y `llevar_en_carretilla`. `Cocinar` requiere `manosLimpias` y produce `cena`. `Envolver` tiene como precondition `silencio` porque es una sorpresa y produce `regalo`. La acción `llevar_a_mano` elimina `basura` pero, el contacto manual con las bolsas, niega `manosLimpias`. La acción final, `llevar_en_carretilla`, también elimina `basura` pero, a causa del desplazamiento ruidoso, niega `silencio`. Inicialmente, tenemos `manosLimpias`, la casa tiene `basura` y está en `silencio`. Las demás proposiciones se consideran falsas.

El grafo de planning se construye de la siguiente manera. Todas las condiciones iniciales son ubicadas en el primer nivel de proposiciones (nivel cero) del grafo. Construir un nivel de acción genérico consiste en lo siguiente. Para cada operador y para cada forma de instanciar las precondiciones de ese operador a proposiciones del nivel previo, se inserta un nodo de acción si no existen dos precondiciones que sean mutuamente exclusivas. Además, se insertan todas las acciones de mantenimiento (acciones nulas) y los arcos de las precondiciones. Luego se chequea la relación de exclusión mutua entre los nodos de acción y se crea una lista, que mantiene esas relaciones, para cada acción. Para crear un nivel de proposiciones genérico, simplemente se tienen en cuenta todos los efectos de las acciones presentes en el nivel previo (incluyendo las acciones de mantenimiento) y se los ubica en el siguiente nivel como proposiciones, conectándolos vía los apropiados arcos de poscondición. Finalmente, dos proposiciones son marcadas como mutuamente exclusivas, si todas las formas de generar la primera son mutuamente exclusivas con respecto a todas las formas de generar la segunda de las proposiciones marcadas.

Como condición necesaria para la existencia de un plan se debe verificar que la totalidad de los literales de la meta estén presentes en el máximo nivel proposicional alcanzado y que ninguno de ellos sea *mutex* con cada uno de los otros. En este caso, la segunda fase de Graphplan es ejecutada: la extracción de la solución.

2.2. Extracción de la Solución

La fase de extracción de la solución se aboca a la tarea de hallar un plan considerando cada una de las n submetas (términos de la conjunción) que conforman la meta. Para cada uno de tales literales presentes en un nivel i , Graphplan elige una acción a en el nivel $i - 1$ tal que produzca esa submeta. Este es un punto de *backtracking*: si más de una acción produce una submeta, entonces Graphplan debe considerar a todas ellas con el fin de asegurar completitud. Si a es consistente (es decir, no *mutex*) con todas las acciones que han sido escogidas hasta ahora en este nivel, entonces se procede a la siguiente submeta. De lo contrario, se realiza backtracking a la elección previa.

Una vez que ha encontrado un conjunto consistente de acciones en el nivel $i - 1$, Graphplan trata recursivamente de encontrar un plan para el conjunto formado a partir de la unión de todas las precondiciones de aquellas acciones, en el nivel $i - 2$. El caso base de la recursión es el nivel cero: si las proposiciones están presentes en ese nivel entonces Graphplan ha encontrado una solución. De lo contrario, si el backtracking falla para todas las combinaciones de las posibles acciones que soportan cada submeta (para cada nivel), entonces Graphplan extiende el grafo de planning con un nuevo nivel de acción y un nuevo nivel de proposición y luego intenta nuevamente la extracción de una solución.

Es importante destacar que Graphplan genera un plan inherentemente paralelo o parcialmente ordenado. Las acciones seleccionadas para el último nivel de acción del ejemplo planteado, `llevar_a_mano` y `envolver` pueden ser ejecutadas en cualquier orden y producirán el mismo efecto. Así, si uno desea una secuencia de acciones totalmente ordenada como plan definitivo, uno podría elegir arbitrariamente: `cocinar`, `llevar_a_mano` y `envolver`.

Una vez obtenido el plan definitivo, surge inmediatamente el siguiente interrogante. ¿Qué sucedería si, una vez ejecutado el plan, el agente descubre que alguna de las acciones necesarias para lograr la meta no produjo los resultados esperados? Ante esta situación sería necesaria una tarea de replaneamiento. El mayor problema, desde un punto de vista computacional, consiste en que el grafo de planning debe volver a construirse desde cero para intentar encontrar nuevamente la solución. En las secciones subsiguientes, consideraremos la conveniencia de adoptar el punto de vista del área de Dinámica de Creencias con el fin de intentar simplificar este problema.

3. Dinámica de Creencias

Dinámica de creencias es el proceso por el cual un agente cambia sus creencias, realizando una transición desde un estado epistémico a otro. Cuando tal agente aprende nueva información puede concluir que esta información contradice sus creencias previas. En este caso, el agente debe revisar sus creencias y decidir cuáles tienen que ser abandonadas con el fin de incorporar la nueva información.

3.1. El Modelo AGM

El modelo AGM[AGM85] constituye una de las aproximaciones más importantes a la formalización de la dinámica de creencias. En este enfoque, los estados epistémicos son representados por conjuntos de creencias que son conjuntos de sentencias cerrados bajo consecuencia lógica.

Notación: Se adopta un lenguaje proposicional \mathcal{L} con un conjunto completo de conectivos booleanos: negación, conjunción, disyunción e implicación. Las fórmulas en \mathcal{L} serán denotadas por letras griegas minúsculas y los conjuntos de sentencias en \mathcal{L} serán denotadas mediante letras mayúsculas. Se emplea un operador de consecuencia Cn . Cn toma un conjunto de sentencias en \mathcal{L} y produce un nuevo conjunto de sentencias. Se asume que el operador Cn satisface las propiedades de *inclusión* ($A \subseteq Cn(A)$), *iteración* ($Cn(A) = Cn(Cn(A))$) y *monotonización* (si $A \subseteq B$ entonces $Cn(A) \subseteq Cn(B)$).

Sea $\mathbf{K} = Cn(\mathbf{K})$ un conjunto de creencias y α una sentencia en un lenguaje proposicional \mathcal{L} . Los tres principales tipos de operaciones de cambio de creencias son los siguientes[Gär88]:

- **Expansión:** Una nueva sentencia es incorporada a un estado epistémico. Si $'+'$ es un operador de expansión entonces $\mathbf{K} + \alpha$ denota el conjunto de creencias \mathbf{K} expandido por α .
- **Contracción:** Alguna sentencia presente en el estado epistémico es retraída sin incorporar nueva información. Si $'-'$ es un operador de contracción entonces $\mathbf{K} - \alpha$ denota el conjunto de creencias \mathbf{K} contraído por α .
- **Revisión:** Una nueva sentencia es incorporada de manera consistente al estado epistémico. Con el fin de hacer posible esta operación, algunas sentencias deben ser retraídas del estado epistémico original. Si $'*'$ es un operador de revisión entonces $\mathbf{K} * \alpha$ denota el conjunto de creencias \mathbf{K} revisado por α .

3.2. Revisión No Priorizada

A partir de los postulados generalmente aceptados para la operación de revisión del modelo AGM, puede deducirse que el sistema está dispuesto a sacrificar cualquier elemento de su teoría con el fin de incorporar la nueva información. Es decir, si la información ya existente es incompatible con la nueva información, esta última tiene preferencia. Por otro lado, una vez que esta nueva información es incorporada, pasa a tener el mismo status y es tan susceptible de ser sacrificada como cualquier otra de la teoría actual.

En [Han96a], Hansson introduce una operación de revisión no priorizada denominada *semi-revision*. Esta operación, denotada “?”, modela la situación en la que un agente recibe nueva información y decide incorporarla o no a su estado de creencias. Este estado epistémico puede estar modelado, básicamente, de dos maneras: mediante *conjuntos de creencias*, que son conjuntos de sentencias clausurados por consecuencia lógica, o mediante *bases de creencias*, que son conjuntos de sentencias no cerrados bajo consecuencia lógica. El uso de bases de creencias parte de la intuición de que algunas de nuestras creencias no tienen sustento independiente, sino que dependen totalmente de creencias más básicas, de las cuales aquellas son inferidas. De este modo, los elementos de una base de creencias representan creencias fundamentales o básicas.

La operación de semi-revisión es caracterizada axiomáticamente a partir de una serie de postulados o criterios de racionalidad. Estos postulados tienen su punto de partida en las propiedades de la operación de revisión del modelo AGM.

En lo que resta de esta sección, nos enfocaremos exclusivamente en la representación de estados epistémicos mediante bases de creencias. Mediante ' $B?\alpha$ ' denotaremos la operación de semi-revisión de una base de creencias B por la sentencia α .

La semi-revisión de bases de creencias puede ser modelada adoptando una versión modificada de la identidad de Levi [Han93]. Es decir, como un procedimiento que consiste de:

1. Adicionar α mediante una operación conjuntista.
2. Hacer consistente la base de creencias, ya sea eliminando α o alguna de las creencias originales.

La segunda de estas operaciones, la de volver consistente una base de creencias, es denominada *consolidation* [Han91]. Un modo posible de llevarla a cabo consiste en realizar una contracción por falso (\perp).

3.2.1. Partial Meet Semi Revision

La implementación más común de la operación de contracción del modelo AGM es *partial meet contraction* [AGM85]. Este proceso consiste en la selección de subconjuntos maximales que no implican la sentencia a contraer y luego, tomar la intersección de todos ellos.

Basada en *partial meet contraction*, es posible definir *partial meet consolidation* como *partial meet contraction* por falso (\perp). La operación de *partial meet consolidation*, denotada ' $!$ ', de un conjunto B es la intersección de los subconjuntos maximales consistentes determinados por una función de selección γ , i.e.,

$$B! = \bigcap \gamma(B \perp \perp).$$

La operación de *partial meet semi revision* puede definirse a partir de la operación de *partial meet consolidation* de la siguiente manera

$$B?\alpha = (B \cup \{\alpha\})! = \bigcap \gamma((B \cup \{\alpha\}) \perp \perp).$$

La ecuación precedente nos indica que, con el fin de realizar una semi-revisión de una base de creencias B por una determinada sentencia α , debemos llevar a cabo la consolidación de la base $B \cup \{\alpha\}$. Esto último consiste en conservar la información común a los subconjuntos maximales consistentes, seleccionados mediante la función de selección γ , de $B \cup \{\alpha\}$.

La operación de *partial meet semi-revision* es caracterizada axiomáticamente de la siguiente manera:

Teorema 3.1. ([Han96a]) Un operador $?$ es un operador de *partial meet semi-revision* si y solo si satisface:

- $A?\alpha$ es consistente (*strong consistency*)
- $A?\alpha \subseteq A \cup \{\alpha\}$ (*inclusion*)

- Si $\beta \in A$ y $\beta \notin A?\alpha$, entonces existe algún A' tal que $A?\alpha \subseteq A' \subseteq A \cup \{\alpha\}$, A' es consistente pero $A' \cup \{\beta\}$ es inconsistente (*relevance*)
- $(A + \alpha)?\alpha = A?\alpha$ (*pre-expansion*)
- Si $\alpha, \beta \in A$, entonces $A?\alpha = A?\beta$ (*internal exchange*)

4. Revisión de Planes

Asumamos que un agente descubre que, una de las acciones requeridas en el plan definitivo no pudo ser ejecutada adecuadamente, es decir, no ha dado los resultados esperados. Ante esta situación, una porción del plan debe ser removida y reemplazada por un subplan conveniente que ofrezca la posibilidad de éxito para el plan global. Sin embargo, esto implica, en el contexto de Graphplan, la reconstrucción del grafo de planning desde el nivel cero.

Es importante tener en cuenta que, la construcción del grafo de planning para un problema determinado no es una tarea trivial. Por lo tanto, es imperativo conservar la mayor parte del grafo ante una modificación del problema original.

Con esta motivación, hemos propuesto la definición de una operación de revisión para grafos de planning. A continuación, resumimos las definiciones necesarias, presentadas en trabajos anteriores, con el fin de introducir el aporte fundamental del presente trabajo. Para mayores detalles, consultar [PS04].

Definición 4.1. Un esquema de acción es una terna (Pre, a, Pos) donde, a es una acción, Pre es un conjunto finito de proposiciones que constituyen las precondiciones de a y Pos es un conjunto finito de proposiciones que se verifican como resultado de aplicar la acción a (poscondiciones de a). \square

Notación: Sea A un esquema de acción. Emplearemos A_{Pre} y A_{Pos} para denotar, respectivamente, las precondiciones y poscondiciones del esquema A . Asimismo, mediante $Met(\Pi)$, denotaremos el conjunto de metas del grafo de planning Π .

Definición 4.2. Sea Π un grafo de planning. Sea n un nodo de proposición o de acción perteneciente al grafo. Mediante la función $Lev_{\Pi}(n)$ se indica el nivel correspondiente al nodo n en el grafo Π . Además, $Lev_{\Pi}(n)$ es indefinido si y sólo si el nodo n no pertenece a Π . \square

Definición 4.3. Sea Π un grafo de planning y sea $A = (Pre, a, Pos)$ un esquema de acción. Decimos que A pertenece a nivel i a Π si y sólo si $Lev_{\Pi}(a) = i$, cada uno de los elementos de los conjuntos Pre y Pos existen como nodos en el grafo y si existen los arcos que modelan las relaciones correspondientes. \square

La tarea básica del operador de revisión es obtener, dado un grafo de planning Π y un esquema de acción A , un nuevo grafo de planning Π' . Este nuevo grafo tiene dos características principales: el esquema A pertenece a nivel i al grafo Π' y, posiblemente, esquemas de acción adicionales han sido removidos de Π' .

La operación de revisión de grafos de planning debería entenderse como un proceso mediante el cual se reemplaza un esquema de acción B por un esquema A que contiene los efectos o poscondiciones de B . Sin embargo, es posible que el esquema a reemplazar y el nuevo esquema difieran en sus precondiciones.

Varios esquemas de acción pueden ser candidatos a ser reemplazados. Una función de selección determina, dentro de este conjunto, el esquema de acción escogido. A continuación, definimos formalmente las nociones de conjuntos de esquemas de acción *pos-equivalentes*, función de selección y revisión de grafos de planning.

Definición 4.4. Sea Π un grafo de planning y sea A un esquema de acción. El conjunto de esquemas de acción *pos-equivalentes* a nivel i con respecto a A , denotado $\Pi\Psi^i A$, se define de la siguiente manera.

$$\Pi\Psi^i A = \{E : E \text{ pertenece al nivel } i \text{ de } \Pi \text{ y } E_{Pos} \subseteq A_{Pos}\}.$$

□

El conjunto de esquemas pos-equivalentes a nivel i con respecto a A , está formado por los esquemas de acción de nivel i cuyas poscondiciones están incluidas en las poscondiciones del esquema de acción A .

Definición 4.5. Sea Π un grafo de planning. Una *función de selección para Π* es una función γ tal que, para cualquier esquema de acción A , se verifica que:

$$\gamma(\Pi\Psi^i A) = \begin{cases} \{E' : E' \in \Pi\Psi^i A\} & \text{si } \Pi\Psi^i A \neq \emptyset \\ \emptyset & \text{si } \Pi\Psi^i A = \emptyset \end{cases}$$

□

La función de selección γ escoge un elemento de $\Pi\Psi^i A$ si el conjunto no es vacío. En caso contrario, la función de selección devuelve el conjunto vacío.

El operador de revisión está definido en base a los operadores de expansión y contracción de grafos de planning. Estos últimos han sido definidos en trabajos anteriores y, por legibilidad, son expuestos nuevamente.

Definición 4.6. ([PS02a]) Sea Π un grafo de planning, $A = (Pre, a, Pos)$ un esquema de acción. La *expansión* a nivel i de Π por A , denotada $\Pi \oplus^i A$, se define como

$$\begin{aligned} \Pi \oplus^i A &= \Pi' = \Pi \cup \{(pre, a) : pre \in Pre\} \cup \{(a, pos) : pos \in Pos\} \\ &\text{donde } Lev_{\Pi'}(pre) = i - 1 \text{ para todo } pre \in Pre, \\ &Lev_{\Pi'}(a) = i \text{ y } Lev_{\Pi'}(pos) = i + 1 \text{ para todo } pos \in Pos \end{aligned}$$

□

Definición 4.7. ([PS01]) Sea Π un grafo de planning, $A = (Pre, a, Pos)$ un esquema de acción. La *contracción* a nivel i de Π por A , denotada $\Pi \ominus^i A$, se define como

$$\begin{aligned} \Pi \ominus^i A &= \Pi' = \Pi - \{(p, a) : p \in Pre \wedge Lev_{\Pi}(a) = i\} - \\ &\{(a, p) : p \in Pos \wedge Lev_{\Pi}(a) = i\}, \text{ tal que } Lev_{\Pi'}(a) \text{ es indefinido} \end{aligned}$$

□

Definición 4.8. Sea Π un grafo de planning, A un esquema de acción y γ una función de selección para Π . La operación de *revisión* de Π con respecto a A , denotada como $\Pi \otimes^i A$, se define como:

$$\Pi \otimes^i A = (\Pi \ominus^i \gamma(\Pi \Psi^i A)) \oplus^i A$$

□

De acuerdo a esta definición, para realizar la revisión a nivel i de un grafo de planning Π con respecto a un esquema de acción A debemos, en primer lugar, contraer a nivel i por el esquema de acción $\gamma(\Pi \Psi^i A)$ y luego, expandir a nivel i por el esquema de acción A .

4.1. Revisión No Priorizada de Planes

A partir de la definición del operador de revisión de grafos de planning y de los postulados propuestos para esta operación[PS02b] es posible inferir que el nuevo esquema de acción tiene preferencia sobre los esquemas de acción ya existentes. Es decir, si en el grafo de planning ya existe un esquema de acción cuyas poscondiciones estén incluidas en el nuevo esquema, este último tiene prioridad sobre el esquema existente. Además, y de manera similar a lo que acontece con el operador de revisión en el modelo AGM, una vez que el nuevo esquema de acción es incorporado, es tan susceptible de ser eliminado como cualquier otro del grafo de planning actual.

La propuesta de este trabajo es introducir una nueva operación de cambio sobre grafos de planning denominada *revisión no priorizada*. Esta operación, denotada ' \odot ', modela la situación en la que un agente planificador recibe un nuevo esquema de acción y decide incorporarlo al grafo de planning, o bien, rechazarlo y conservar los esquemas de acción originales.

Analicemos un ejemplo concreto tomando como base el *dinner-date problem*¹(Figura 1). Supongamos que, una vez construido el plan definitivo, el agente decide revisar el grafo de planning obtenido, a nivel 1, por el esquema compuesto por la acción `llamar_rotiseria` con efecto `cena` y precondition `dinero`. Como resultado de la revisión, el esquema de acción (`{ManosLimpias}`, `cocinar`, `{cena}`) sería eliminado del grafo de planning en favor del esquema por el cual se revisa el grafo. La operación de revisión no priorizada otorga, en cambio, un grado de flexibilidad adicional: permite al agente planificador, además, rechazar el nuevo esquema propuesto y conservar los esquemas de acción originales.

Dado un grafo de planning Π y un esquema de acción A , la revisión no priorizada de Π por A puede ser modelada como un procedimiento que consiste de:

1. Realizar una expansión de Π por A .
2. *Consolidar* el grafo de planning resultante, ya sea eliminando A o alguno de los esquemas originales.

A continuación, definimos formalmente la operación de revisión no priorizada de grafos de planning.

¹En realidad, consideramos una variante del *dinner-date problem* donde se agrega la proposición `dinero` a las condiciones iniciales.

Definición 4.9. Sea Π un grafo de planning, A un esquema de acción y γ una función de selección para Π . La operación de *revisión no priorizada* de Π con respecto a A , denotada como $\Pi \odot^i A$, se define como:

$$\Pi \odot^i A = (\Pi \oplus^i A) \ominus^i \gamma((\Pi \oplus^i A) \Psi^i A)$$

□

De acuerdo a esta definición, para realizar la revisión no priorizada a nivel i de un grafo de planning Π con respecto a un esquema de acción A debemos, en primer lugar, expandir a nivel i por el esquema de acción A y luego, contraer a nivel i por el esquema de acción $\gamma((\Pi \oplus^i A) \Psi^i A)$. Teniendo en cuenta la definición de la función γ (definición 4.5) es claro que el esquema seleccionado para eliminar podría ser alguno de los esquemas originales o el nuevo esquema propuesto. Considerando el ejemplo planteado, la función de selección γ podría seleccionar para eliminar al esquema con acción `llamar_rotiseria`, o bien, al esquema con acción `cocinar`. Ninguno de los dos esquemas tiene, en principio, preferencia sobre el otro.

La característica fundamental del operador de revisión no priorizada de grafos de planning es la no satisfacción del postulado de éxito para revisiones. Las propiedades adicionales del operador son sintetizadas mediante el siguiente resultado.

Teorema 4.2. Sea Π un grafo de planning y sean A y B dos esquemas de acción. Sea \odot^i un operador de revisión no priorizada de acuerdo a la definición 4.7. Entonces \odot^i satisface las siguientes propiedades:

- *Inclusión:* $(\Pi \odot^i A) \subseteq (\Pi \oplus^i A)$
- *Expansión Previa:* $(\Pi \oplus^i A) \odot^i A = \Pi \odot^i A$
- *Intercambio Interno:* Si A y B pertenecen a nivel i a Π , entonces $(\Pi \odot^i A) = (\Pi \odot^i B)$
- *Completitud:* $Met(\Pi) = Met(\Pi \odot^i A)$

Las propiedades de *inclusión*, *expansión previa* e *intercambio interno* son consecuencia directa de la definición de las operaciones de revisión no priorizada y expansión de grafos de planning. *Completitud* es satisfecha dado que es imposible que desaparezcan metas del grafo de planning luego de realizar una revisión no priorizada por un esquema de acción determinado.

5. Conclusiones

En este trabajo, hemos presentado la operación de revisión no priorizada de grafos de planning. El operador ha sido definido de manera constructiva y hemos analizado sus propiedades. La característica fundamental del operador de revisión no priorizada es la no satisfacción del postulado de éxito para revisiones. Esto nos brinda un grado de flexibilidad adicional dado que el nuevo esquema de acción no tiene prioridad sobre los esquemas ya existentes. Un agente planificador, que recibe propuestas de nuevas piezas para el plan, tiene ahora la facultad de decidir su rechazo o su incorporación al grafo de planning definitivo.

Referencias

- [AGM85] Carlos Alchourrón, Peter Gärdenfors, and David Makinson. On the Logic of Theory Change: Partial Meet Contraction and Revision Functions. *Journal of Symbolic Logic*, (50):510–530, 1985.
- [BF95] A. Blum and M. Furst. Fast planning through planning graph analysis. In *Proceedings of the XIV International Joint Conference of AI*, pages 1636–1642, 1995.
- [BF97] A. Blum and M. Furst. Fast planning through planning graph analysis. *J. Artificial Intelligence*, (90):281–300, 1997.
- [FN71] Richard Fikes and Nils Nilsson. STRIPS: A new approach to the application of theorem proving to problem solving. *J. Artificial Intelligence*, (2), 1971.
- [Gär88] Peter Gärdenfors. *Knowledge in Flux: Modelling the Dynamics of Epistemic States*. MIT Press, Cambridge, England, 1988.
- [GPP⁺99] M. Georgeff, B. Pell, M. Pollack, M. Tambe, and M. Wooldridge. The Belief-Desire-Intention Model of Agency. In M.P.Singh J.P.Müller and A.S. Rao, editors, *Intelligent Agents V*, volume 1555 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag, Berlin, Germany, 1999.
- [Han91] Sven Owe Hansson. *Belief Base Dynamics*. PhD thesis, Uppsala University, Uppsala, Sweden, 1991.
- [Han93] Sven Owe Hansson. Reversing the Levi Identity. *Journal of Philosophical Logic*, (22):637–669, 1993.
- [Han96a] Sven Owe Hansson. Semi-Revision. *Journal of Applied Non-Classical Logic*, 1996.
- [Han96b] Sven Owe Hansson. *A Textbook of Belief Dynamics*. Kluwer Academic Press, 1996.
- [PS01] G. Parra and G. Simari. Replaneamiento en Agentes Inteligentes. Contracción de Grafos de Planning. In *VII Congreso Argentino de Ciencias de la Computación*, volume 2, pages 1081–1093, Universidad Nacional de la Patagonia Austral - El Calafate - Santa Cruz, 2001.
- [PS02a] G. Parra and G. Simari. Reelaboración de Planes en Agentes Inteligentes. Expansión de Grafos de Planning. In *Jornadas Chilenas de Computación 2002*, Universidad de Atacama - Copiapó - Chile, 2002.
- [PS02b] G. Parra and G. Simari. Replaneamiento en Agentes Inteligentes. Revisión de Grafos de Planning. In *VIII Congreso Argentino de Ciencias de la Computación*, Buenos Aires, 2002.
- [PS04] G. Parra and G. Simari. Funciones de Selección Cualitativas y Cuantitativas para la Revisión de Planes en Agentes Inteligentes. In *X Congreso Argentino de Ciencias de la Computación*, Universidad Nacional de La Matanza - San Justo - Buenos Aires - Argentina, 2004.
- [Wel99] Daniel S. Weld. Recent Advances in AI Planning. *AI Magazine*, 1999.